(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 October 2001 (04.10.2001)

PCT

(10) International Publication Number
WO 01/73522 A2

(51) International Patent Classification⁷: G06F

(21) International Application Number: PCT/US01/40272

(22) International Filing Date: 9 March 2001 (09.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/538,860 29 March 2000 (29.03.2000) US

(71) Applicant: NETFISH TECHNOLOGIES, INC. [US/US]; Suite 650, 2350 Mission College Blvd., Santa Clara, CA 95054 (US).

(72) Inventor: BALABINE, Igor; 11063 Bel Aire Court, Cupertino, CA 95014 (US).

(74) Agents: BEAMER, Norman, H. et al.; Fish & Neave, 1251 Avenue of the Americas, New York, NY 10020 (US).

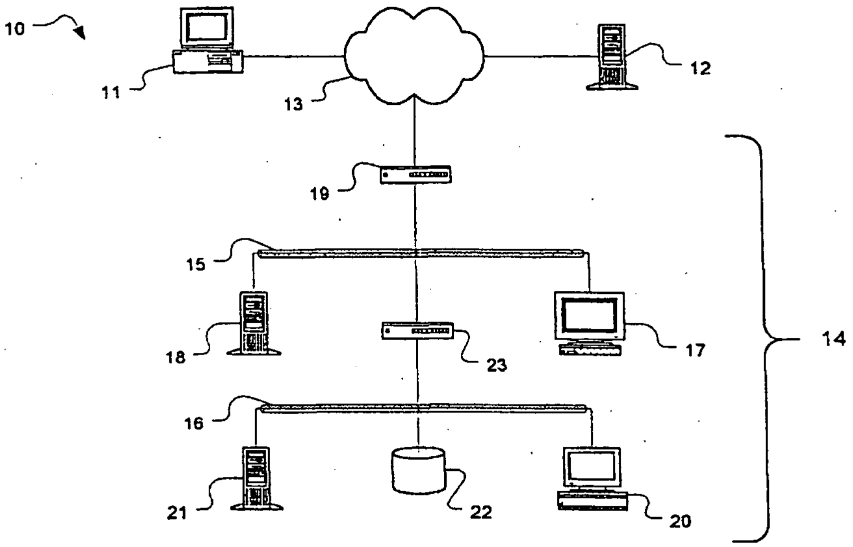
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHODS AND APPARATUS FOR SECURING ACCESS TO A COMPUTER



(57) Abstract: Methods and apparatus are provided for providing secure communications around a firewall. The firewall is configured to only pass packets if they originate on a more protected side of the firewall, of if they are in response to such a packet. A persistent secure connection is made from a connection manager program inside the firewall to a server outside the firewall. A protocol request message is then sent from the connection manager to the server over the secure connection. Requests arriving from a client are embedded in a protocol response message and sent to the connection manager program. After processing, the results, if any, are sent as a protocol request message to the server, which extracts the result from the protocol request message and puts the result in a protocol response message which is sent to the requesting client.

WO 01/73522 A2

METHODS AND APPARATUS FOR SECURING ACCESS TO A COMPUTER

Field of the Invention

The present invention relates generally to methods and apparatus for providing secure remote access to a computer, and more specifically to providing secure access to a computer behind a firewall.

Background of the Invention

The Internet is a vast, globe-spanning, collection of interconnected computer networks and the associated programs, protocols, and standards that enable these computers to communicate with each other. The World Wide Web ("web"), a popular application of the Internet, relies on some of these protocols and standards to make vast collections of digital content accessible via the Internet. Because basic web technologies are relatively simple, anyone may readily publish content ranging from simple text to demanding multi-media presentations.

The globe-spanning nature of the Internet lets a user contact any computer connected to the Internet from any other computer connected to the

Internet. This fundamental property of the Internet, combined with the ease of publishing content on the web, is largely responsible for the explosive growth of the Internet as a medium of communication.

An unfortunate side effect of the globe-spanning nature of the Internet is that any computer connected to the Internet may be a target for attack by meddlesome individuals from anywhere in the world. Indeed, there are countless reports of individuals gaining unauthorized access to computers and other devices connected to the Internet. Conventional wisdom has, therefore, been that sensitive, proprietary, or confidential information should not be stored on computers connected to the Internet. To do otherwise may expose these computers to outside attack and risk compromising any sensitive data they contained.

However, because of the rapid growth in business-to-business electronic commerce, it is often desirable or even necessary to be able to share sensitive data via the Internet. For example, a company may need to share financial projections with potential investors, or a manufacturing partner may need design specifications for a new product. One way to protect such sensitive data is to use a firewall to restrict access to the computers storing the sensitive data.

A firewall is a combination of hardware and/or software that serves as a controlled link between one network, such as the Internet, and a protected network, such as a corporate Intranet. As used herein, a network referred to a "protected," or as being "inside" or "behind" a firewall, refers to a network that is being protected by the firewall. Conversely, a network referred to as "unprotected," or "less-protected," or as being "outside" a firewall, refers to a network that is not being protected by the firewall. For instance, a corporate Intranet is generally behind a firewall, whereas the Internet is outside the firewall.

Generally, a firewall examines packets arriving at the firewall and processes the packets according to a set of rules and policies. For example, a firewall may implement a policy of forwarding packets that originate behind the firewall, but to deny or drop packets originating from outside the firewall. Rules may then provide for limited exceptions to the more general policies. A common rule is to allow a packet originating outside the firewall to pass through if it is a response to a packet that originated within the protected network. A rule such as this is necessary for the Internet services that use the Transmission Control Protocol (TCP), because TCP requires a receiving computer to send acknowledgments of data that has been received.

Another common rule is to allow packets from specific IP addresses or IP domains to pass through the fire wall. Such a rule may be used, for example, to provide trusted individuals with access to proprietary data stored on a computer behind a firewall. However, this type of rule is, in a sense, a small hole in a firewall that lets packets pass through. Such holes are potential weak spots that may be exploited to gain unauthorized access to a computer and to any confidential information it may contain. For example, the originating IP address of a packet may be forged using a technique known as source spoofing to make a packet appear to come from a 'friendly' IP address or domain when in fact the packet originated elsewhere. Source spoofing is only one form of attack that may be used in an attempt to gain unauthorized access; many other methods of exploiting weaknesses in firewall security are known in the art, and new forms of attack are continually being discovered.

In view of the forgoing it would, therefore, be desirable to provide methods and apparatus for securing a firewall against common forms of attack.

It would also be desirable to provide methods and apparatus for allowing packets to pass through a firewall without weakening firewall protection.

In addition, it would be desirable to provide methods and apparatus for providing secure access through a firewall.

Summary of the Invention

It is, therefore, an object of the present invention to provide methods and apparatus for securing a firewall against common forms of attack.

It is also an object of the invention to provide methods and apparatus for allowing packets to pass through a firewall without weakening firewall protection.

It is another object of the invention to provide methods and apparatus for providing secure access through a firewall.

These and other objects of the present invention are achieved by providing a multiplexer and a connection manager. The connection manager, located behind a firewall, establishes an outgoing connection to the multiplexer and sends the multiplexer a request message. The multiplexer, which is located outside the firewall, receives and queues the request message, keeping the connection open.

A server outside the firewall receives an request from a client and forwards it to the multiplexer. The multiplexer dequeues the previously queued request message and creates a response message containing the client request. The response message, including the client request, is then sent to the connection manager.

The connection manager removes the client request from the response message and sends it to a protected application, or back-end, for processing. When the processing is complete, the connection manager sends the back-end response to the multiplexer in another request message. The multiplexer removes the response from the request message and passes the response to the outside server for sending to the requesting client.

As a result of this process, all packets passing through the firewall originate behind the firewall, or are responses to packets originating behind the firewall.

Brief Description of the Drawings

The above and other objects and advantages of the present invention will be apparent upon consideration of the following detailed description taken in conjunction with the accompanying drawings, in which like characters refer to like parts throughout, and in which:

FIG. 1 is a simplified diagram of wide area network, including a hierarchical computer network connected thereto;

FIGS. 2A and 2B show the process of initializing the present invention;

FIGS. 3A and 3B shows how client requests are passed from a less-protected network to a more-protected network; and

FIGS. 4A and 4B show how a response is returned from the more-protected network to the client on the less-protected network.

Detailed Description of the Invention

The web comprises many interconnected computers and networks, such as personal computer 11, web server 12, and network 13 of FIG. 1. Personal computer 11 may comprise any general purpose computer such as an IBM PC compatible computer, an Apple PowerMac, a Sun workstation, or a dumb network terminal. Web server 12 typically comprises a high performance computer designed specifically to serve as a network server, however in some situations a computer similar to personal computer 11 may also be used. The computers are interconnected by network 13 which is preferably the Internet, but may be a local area network (LAN), a wide area network (WAN), a virtual private network (VPN), or other type of network.

Also shown in FIG. 1 is private network 14 comprising LANs 15 and 16 which may be, for example, a corporate Intranet comprising Ethernet-type networks. Workstation 17 and server 18 are connected to LAN 15 which is in turn is connected to network 13 by outer bridge 19. Outer bridge 19, which may also be a router, switch, or other similar device, may implement a firewall to provide some protection to workstation 17, server 18, and any other device connected to LAN 15.

Workstation 20, server 21, and database 22 are connected to secure LAN 16, which is in turn connected to LAN 15 by bridge 23 . Bridge 23 implements a firewall to provide increases protection to LAN 16 and the devices connected to LAN 16. By appropriately configuring the firewall on bridge 20, access to LAN 16 may be made more restrictive than access to LAN 15. For instance, bridge 20 may be configured with a policy of denying all packets originating outside bridge 20, but with a rule to let through packets originating from selected IP addresses. Such a configuration provides a range of security as may be appropriate for the computers attached to LAN 16. Thus, sensitive data should be kept on server 21 attached to LAN 16 and less sensitive, or public, information may be kept on server 18 attached to LAN 15.

However, as discussed in the background of the invention, allowing even limited access to LAN 16 through the bridge 20 firewall may provide a potential weakness that might be exploited to attack the firewall and gain access to LAN 16. Therefore, in accordance with the principles of the present invention, no packets originating from outside the bridge 20 firewall are permitted to pass through the firewall onto LAN 16.

Referring now to FIGS. 2A and 2B, operation of a preferred embodiment of the present invention is described in more detail. As shown in FIG. 2A, the present invention comprises two components located on either side of firewall 25. The first component, gateway 26 comprises web server 27 and multiplexer 28 and runs on a computer or server outside of firewall 25. The second component, comprising connection manager 29 runs on a computer or server behind firewall 25. For instance, gateway 26 and connection manager 29 may be run, respectively, on servers 18 and 21 of FIG. 1. Also shown is back-end 31 which comprises one or more programs for providing services such as searching database 22 of FIG. 1.

The principles of the present invention are disclosed herein in terms of the components and arrangement shown in FIG. 1. However, one skilled in the art will understand that various other programs or modules may be used instead of, or in addition to, back end server 31, that the various components of gateway 26 may be run on the same or different computers, and that connection manager 29 and back-end 31 need not be run on the same computer. Furthermore, the description is made in terms of specific Internet protocols, which the skilled artisan will understand are selected for purposes of illustration only and that other protocols may be substituted therefor.

Gateway web server 27 is a conventional web server that supports the http or https schemes, such as the Microsoft Internet Information Server, available from Microsoft Corporation, Redmond, Washington, or the open source web server Apache, available from http://www.apache.org. Gateway web server 27 is configured to receive and process https requests received at certain specified URLs. Of these specified URLs, selected ones are configured to only accept connections from connection manager 29. In addition, firewall 25 is configured to only pass packets originating on secure LAN 16, and responses to such packets if they originate from web server 27. These configurations of gateway web server 27 and firewall 25 provide improved protection for LAN 16.

In accordance with the principles of the present invention, connection manager 29 establishes a pool of TCP/IP connections with gateway web server 27 for use in communicating across firewall 25. A process for establishing the connection pool is shown in FIG. 2B. To establish the pool of connections, at steps 35 and 36 connection manager 29 establishes a TCP/IP connection with web server 25. Preferably, the connection is secured using technologies such as secure socket layer (SSL). At step 37 http request message 32 is sent to web server 25 over the established TCP/IP connection.

```
POST <GW_URL> HTTP/1.1
Connection: keep-alive
Pragma: msgid=”0"
Content-Length: 0
CRLF
```

Listing **1**

An exemplary http request message is shown in Listing 1. The exemplary http request message is a POST type message conforming to the http protocol and sent to the Uniform Resource Locator (URL) identified by <GW_URL>, which is one of the specified URLs configured to only accept connections from connection manager 29. The header line Connection: keep-alive specifies that gateway web server 27 should keep the TCP/IP connection open after responding to the initial request. Pragma: msgid = "0" identifies the http request message as a setup message, Content-Length: 0 indicates that there is no body to the http request message, and CRLF is a blank line terminating the header of the http request message.

At step 39 web server 27 accepts the https request message from connection manager 29, starts a new thread of execution, and sends the request to multiplexer 28 for processing. At step 40, multiplexer 28 stores the request in a queue, associates a wake-up event with the current thread, and suspends the tread until the wake-up event is triggered. Gateway web server 27 then waits for incoming client requests at step 41, while, at step 38, connection manager 29 waits for a response to https request message 32.

Referring now to FIGS. 3A and 3B, at step 45, gateway web server 27 receives client request message 46 at the service URL. Web server 27 invokes an appropriate thread of execution which passes the client request message to multiplexer 28. At step 47, multiplexer 28 builds an http *response* message containing as its body client request message 46. Multiplexer 28 dequeues, at step 48, an https request queued at step 40 of FIG. 2B and wakes up the associated thread, passing it the https response message containing client request message 46. Multiplexer 28 returns the response message to web server 27 which sends, at step 50, https response message 49 to connection manager 29 as a response to https request message 32 of FIG. 2A.

An exemplary https response message is shown in Listing 2. The header line `Connection: keep-alive` indicates that the underlying TCP/IP connection is to remain open. *`<Message content length>`* is the length of client request message 46 embedded in the https response message, and <*N*> is a unique transaction identification number for identifying the client request. *`<Client message>`* is a copy of client request message 46.

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: <Message content length>
Pragma: msgid = "<N>"
MIME-Version: 1.0
Content-type: text/html
CRLF
<Client message>
CRLF
```

Listing 2

At step 51, connection manager 29 receives https response message 49 and extracts http client request message 46', saving the transaction identification number for later use. At step 52, http request message 46' is sent to back-end 31 for processing as required. Connection manager 29 also sends, at step 54, another https setup request message, similar to https setup request message 32, back to web server 27 to maintain the pool of TCP/IP connections and gateway web server 27 resumes waiting for another client request message at step 55.

The remainder of the method of the present invention is described with reference to FIGS. 4A and 4B. When finished processing http client request 46', back end 31 sends results 60 to connection manager 29. At steps 62 through 64, connection manager 29 encapsulates results in https request message 65 and sends it to web server 27. An exemplary http request message is shown in Listing 3, wherein <*N*> is the transaction identification number saved in step 51, `<Back-end results>` comprises back-end response message 60, and `<Response message length>` is the length thereof.

```
POST <GW_URL> HTTP/1.1
Connection: keep-alive
Pragma: msgid = "<N>"
Content-Length: <Response message length>
CRLF
<Back-end results>
CRLF
```

Listing 3

At step 66, gateway web server 27 receives http request message 65 and sends it to multiplexer 28. Multiplexer 28, at step 67 retrieves the transaction id and back-end results 60' from https request message 65. The transaction id is used to associate back-end result 60' with original http client request 46 of FIG. 3A. At step 68, multiplexer 28 sends server response 60' to web server 27 which sends it to the client in http client response message 69.

Multiplexer 28 also sends, via web server 27, at step 71, confirmation message 73 to indicate that http client response message 69 has been sent to the client. Gateway web server 27 and multiplexer 28 then wait, at step 74 for the next http client request. At step 75, connection manager 29 notifies back-end 31 that the response was delivered thereby signaling the end of the transaction. At step 76, connection manager 29 keeps the TCP/IP connection to gateway web server 27 open, waiting for the next https response containing a client request.

It will be appreciated by one skilled in the art that while a preferred illustrative embodiment of the present invention is described above, the present invention may be practiced by other than the described embodiment, which is presented for purposes of illustration and not of limitation, that various changes and modifications may be made to the illustrative embodiment disclosed herein, and that the present invention is limited only by the following claims which are intended to cover all such changes and modifications which may fall within the true spirit and scope of the invention.

What Is Claimed Is:

1. A method of providing secure access to a protected resource on a computer network, wherein the computer network includes a first server and a firewall, and wherein the first server is outside the firewall and the protected resource is inside the firewall, the method comprising:

accepting at the first server an initial request from the protected resource;

accepting at the first server a client request requiring access the protected resource;

sending from the first server to the protected resource a response to the initial request, the response including the client request;

accepting, at the first server, a second request from the protected resource, the second request including a reply to the client request; and

sending the reply to the client from the first server.

2. The method of claim 1 wherein accepting the initial request and sending the response comprise using a request-response protocol.

3. The method of claim 2 wherein using a request-response protocol comprises using the hypertext transfer protocol.

4. The method of claim 2 wherein using a request-response

protocol comprises using a secure protocol.

5. The method of claim 4 wherein using a secure protocol comprises using the Secure Sockets Layer protocol.

6. The method of claim 1 wherein first server comprises a web server, the method further comprising processing the client request to build the response to the initial request.

7. The method of claim 6 wherein processing the client request comprises executing a routine on the web server.

8. A method of providing secure access to a protected resource on a computer network, wherein the computer network includes a first server and a firewall, and wherein the first server is outside the firewall and the protected resource is inside the firewall, the method comprising:

sending from the protected resource an initial request to the first server;

accepting from the server a response to the initial request, wherein the initial response includes a client request;

processing the client request to generate a reply thereto;

sending a second request to the server, wherein the second request includes the reply to the client request.

9. The method of claim 8 wherein sending the initial request and accepting the response comprise using a request-response protocol.

10. The method of claim 9 wherein using a request-response protocol comprises using the hypertext transfer protocol.

11. The method of claim 9 wherein using a request-response protocol comprises using a secure protocol.

12. The method of claim 11 wherein using a secure protocol comprises using the Secure Sockets Layer protocol.

13. The method of claim 8 wherein processing the client request comprises sending the client request to a back-end server.

14. The method of claim 8 wherein the protected resource includes a web server and wherein processing the client request comprises executing a programmed routine on the web server.

15. The method of claim 14 wherein executing a routine on the web server comprises running a servlet on the web server.

16. Apparatus for providing secure access to a protected resource,

wherein the protected resource is coupled to a computer network inside a firewall, the apparatus comprising:

a first computer located outside the firewall; and

a second computer located inside the firewall;

wherein:

the first computer is programmed to:

accept an initial request from the second computer;

accept a client request requiring access the protected resource;

send to the second computer a response to the initial request, wherein the response includes the client request;

accept a second request from second computer,. the second request including a reply to the client request; and

send the reply to the client;

and the second computer is programmed to:

send the initial request to the first computer;

accept the response to the initial request;

process the client request to generate a reply thereto; and

send the second request to the first computer.

17. The apparatus of claim 16 wherein the first and second computers are programmed to use a request-response protocol.

18. The apparatus of claim 17 wherein the first and second

computers are programmed to use the hypertext transfer protocol.

19. The apparatus of claim 17 wherein the first and second computers are programmed to use a secure protocol.

20. The apparatus of claim 19 wherein the first and second computers are programmed to use the Secure Sockets Layer protocol.

21. The apparatus of claim 16 wherein first computer is programmed with routines implementing a web server.

22. The apparatus of claim 21 wherein being programmed to process the client request comprises being programmed with a servlet for processing the client request.

23. The apparatus of claim 21 wherein being programmed to process the client request comprises being programmed to send the client request to the protected resource and receiving a reply from the protected resource.

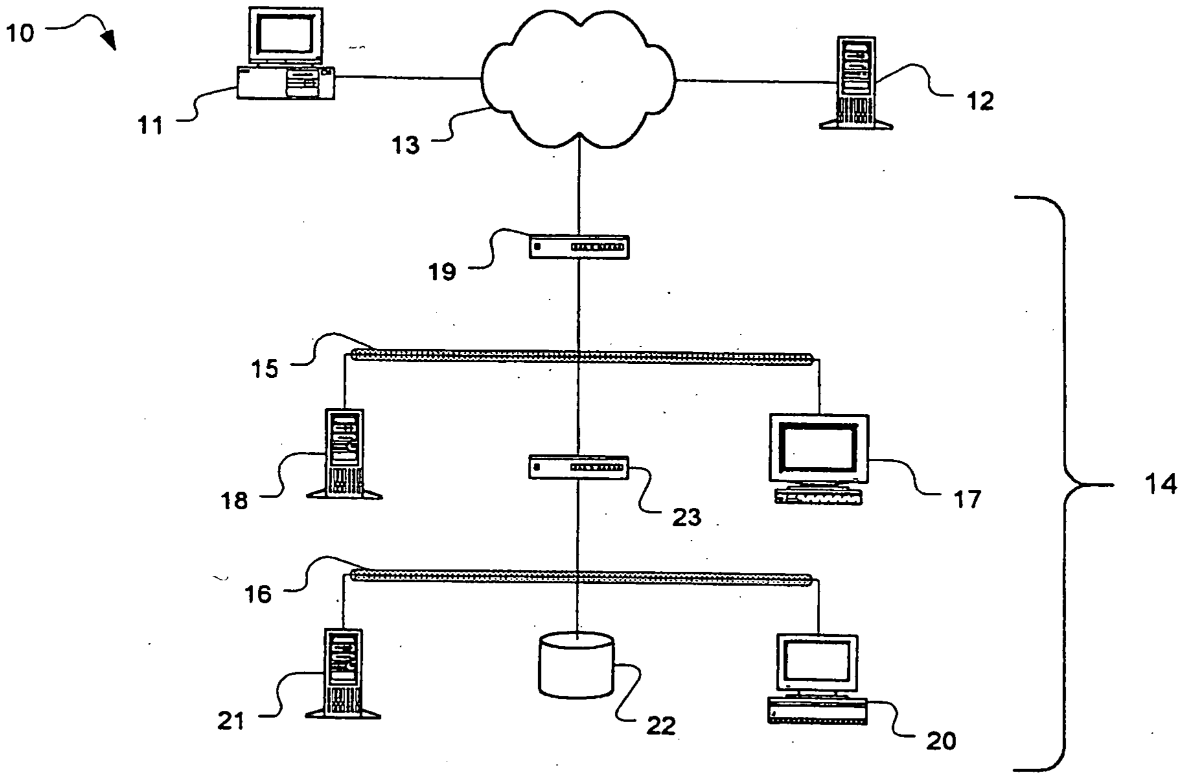


FIG. 1

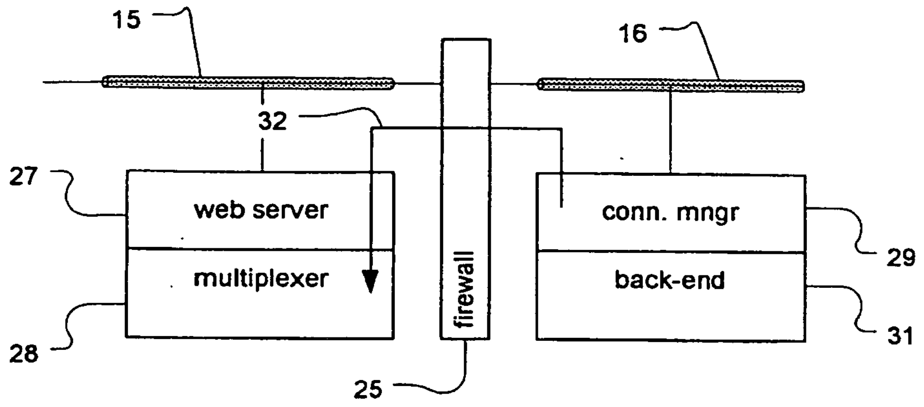
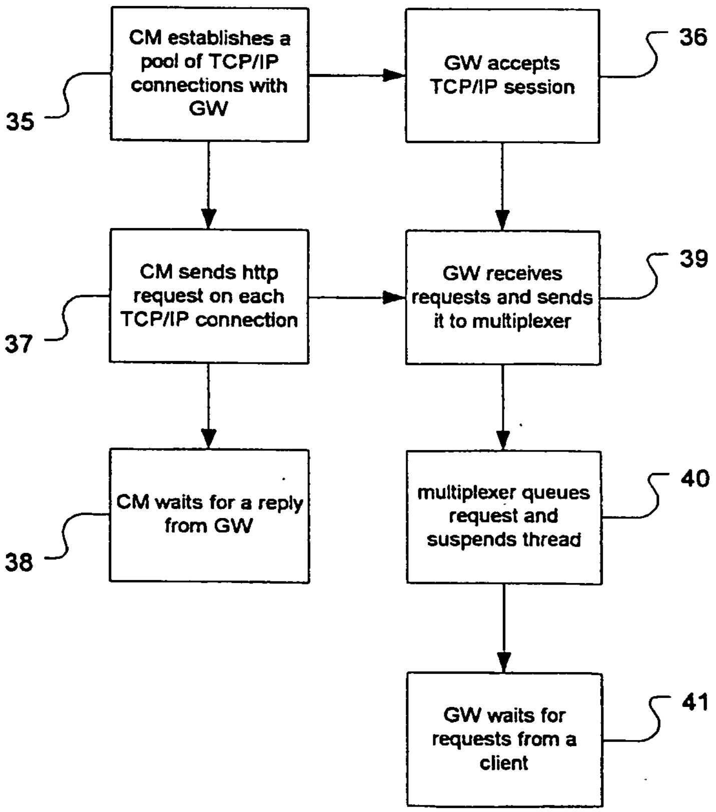

15
16
32
27
web server
multiplexer
28
firewall
25
conn. mngr
back-end
29
31

FIG. 2A

FIG. 2B


CM establishes a pool of TCP/IP connections with GW
35
GW accepts TCP/IP session
36
CM sends http request on each TCP/IP connection
37
GW receives requests and sends it to multiplexer
39
CM waits for a reply from GW
38
multiplexer queues request and suspends thread
40
GW waits for requests from a client
41

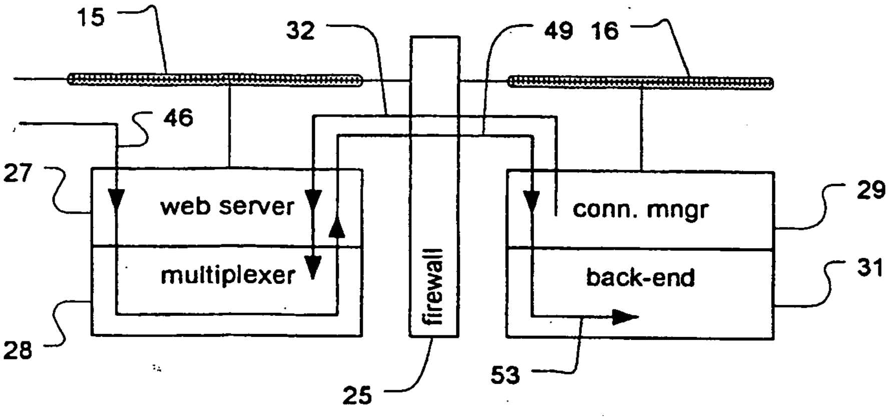

15
32
49
16
46
27
web server
multiplexer
firewall
conn. mngr
back-end
29
31
28
25
53

FIG. 3A

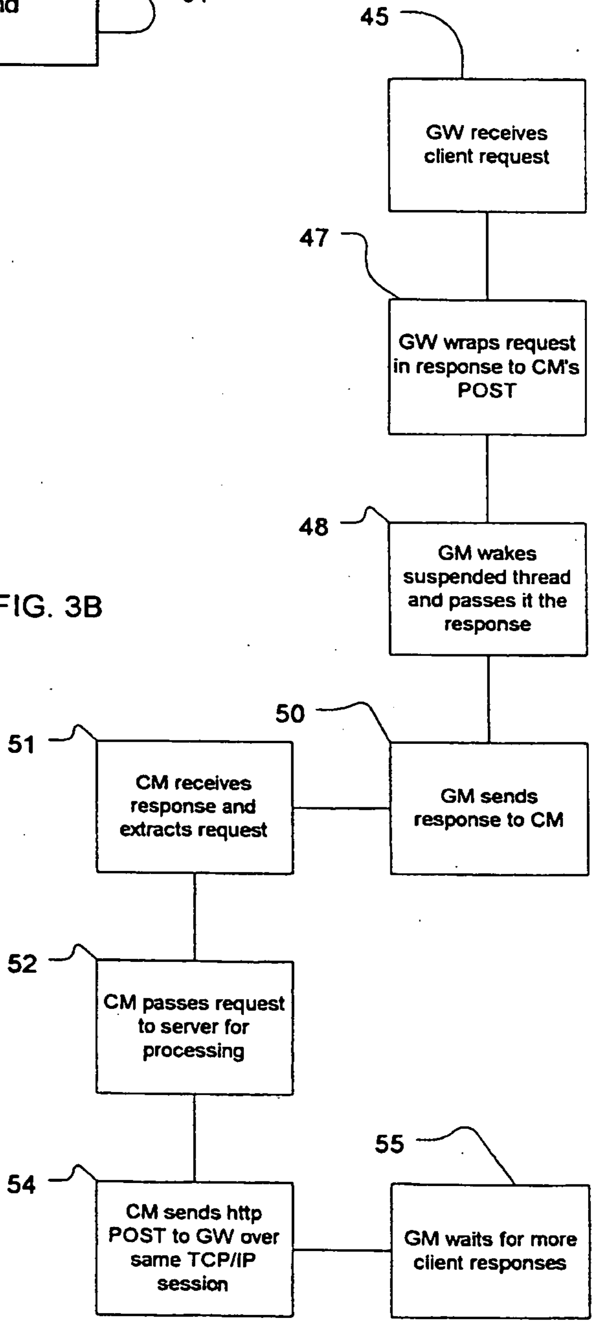

45
GW receives client request
47
GW wraps request in response to CM's POST
48
GM wakes suspended thread and passes it the response
50
GM sends response to CM
51
CM receives response and extracts request
52
CM passes request to server for processing
54
CM sends http POST to GW over same TCP/IP session
55
GM waits for more client responses

FIG. 3B

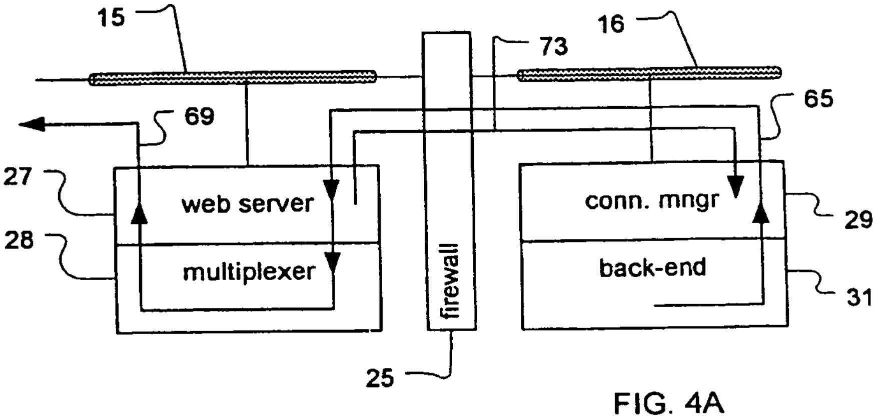

15
16
73
65
69
27
web server
28
multiplexer
firewall
conn. mngr
29
back-end
31
25

FIG. 4A

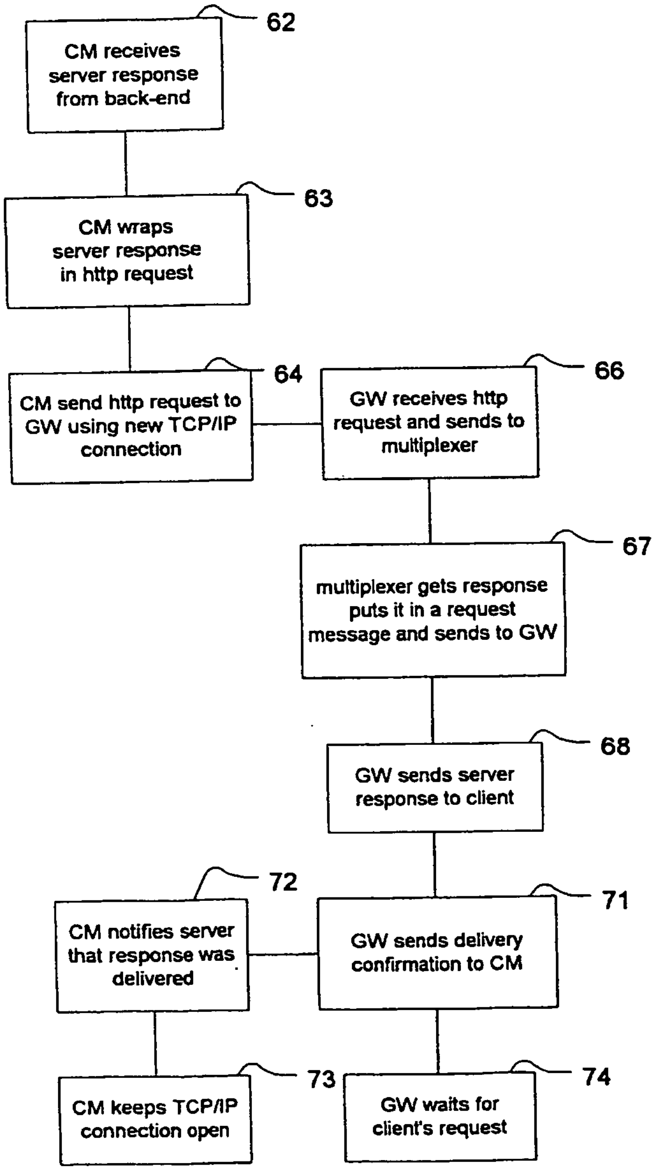

CM receives server response from back-end
62
CM wraps server response in http request
63
CM send http request to GW using new TCP/IP connection
64
GW receives http request and sends to multiplexer
66
multiplexer gets response puts it in a request message and sends to GW
67
GW sends server response to client
68
CM notifies server that response was delivered
72
GW sends delivery confirmation to CM
71
CM keeps TCP/IP connection open
73
GW waits for client's request
74

FIG. 4B